Detecting Latent Variables of Interest in Geo-localized Environments Using an Aerial Robot

David Saldaña[1], Ramon Melo[1], Erickson R. Nascimento[1], and Mario F. M. Campos[1]

***Abstract*— In general, monitoring applications require human intervention whenever there is no physical sensors for the variables of interest (*e.g.* people in danger after a catastrophe). In this paper we describe an inference engine which is used to estimate latent variables that can not be perceived by sampling the physical phenomena directly. Our approach uses information from different types of sensors, and fuses them along with knowledge of experts. The inference engine works with probabilistic first order logic rules based on geo-located sensed data as evidences in order to dynamically create the structure of a *Bayesian network*. Our experiments, performed by using an aerial robot with a mounted RGB-Camera, show the capability of our method to detect people in danger situations, where the physical variables to being sensed are humans and fire.**

***Index Terms*— Inference engine, Information Fusion, Bayesian Networks, Predictive Situation Awareness.**

I. Introduction

There are a variety of environments where monitoring actively is a priority, since most catastrophes begin with small events without early identification. Examples of these kind of environments are: minefields, forests, and devastated cities. In those scenarios, most of the actions must be executed by humans, since some variables are not easy to identify in an autonomous way. However the state of the art methods points towards to create autonomous systems with capabilities to detect events of interest, that cannot be sensed directly, such as: flaming objects, broken pipes, spilled water, trapped animals, source of hazardous chemical leaks or lying people on the floor.

While conventional static sensor networks have difficulties in this context, since covering the full environment requires to spread a number of sensors everywhere. Sensors like RGB Cameras are widely used because they have a good cost/benefit, provide a wide visual angle, high resolution and excellent quality. By the same way, some sensors can be very accurate, but they are not light enough to be carry on a small aerial vehicle. In the aforementioned cases, some situations cannot be recognized only using the data acquired by sensors, the available technologies do not offer sensors with full certainty and most of the times a sensor fusion is required.

[1]D. Saldaña,R. R. Melo, E. R. Nascimento and M.F.M. Campos are with the Computer Vision and Robotics Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, MG, Brazil. E-mails: {saldana, ramonmelo, erickson, mario }@dcc.ufmg.br

*The authors also gratefully acknowledge the support of CAPES, CNPq and FAPEMIG.

Under the stated premises, the use of mobile sensors offers a promising solution in those contexts, on account of using multiple mobile sensors equipped with different kind of sensor can be more efficient than a single sensor. Additionally, the distributed information can be fused to improve estimations about interested objects and events in the environment.

In this work we present a method for inference and information fusion in partially observable environments, where there is only information about the map boundaries and sensed data. It is focus on fuse sensed data and the expert's knowledge in order to identify regions of interest, based on evidences and a set of rules defined by an expert. We present a generic framework that can be applied to scenarios where variables of interest cannot be sensed directly. In this work, a situational awareness after a catastrophe is modeled, where there are people lying on the floor and objects on fire, and then, we detect, and classify them by type, and severity.

The remaining of this article is presented as follows: section II present a summary of related works in the research area; our problem model, and all information needed to perform the inference evaluation is described in section III; section IV describes the *inference engine* for information fusion and inference. Section VI describes the experiments executed in order to validate the engine; finally, conclusions on section VII.

II. Related works

An earlier work deploying mobile sensors in the environment is shown in [1]. It describes the basic requirements for a robotic system being capable to locate a chemical leak. The use of a system like this can perform detection on extreme conditions of radiation and/or toxic chemical concentration, that would be impossible the accomplishment by humans.

The work of Drews et al. [2] integrates multiple visual static sensor information with probabilistic graphical models. This integration allows to infer crowd behaviors with a probabilistic method based on provided information by multiple RGB and IR cameras looking at the same scene but from different points of view. The fusion technique uses *Hidden Markov Networks (HMM)* [3] and *Bayesian Networks* [4] with different results for each one, but both were able to detect abnormal crowd behaviors. However the fusion method requires multiple and redundant sensors looking for the same place at the same time, which is a hard constrain for large environments.

The work [5] applies *Partially Observable Markov Decision Process (POMDS)* for cooperative active perception.

Mobile robots and static sensors work autonomously together to identify flames or persons who need of assistance such as: waving, running, or lying on the floor. It is implemented on a group of static cameras along with a mobile robot to work autonomously and cooperatively. The main drawback of this approach is the lack of scalability because the number of states increases exponentially with the number of agents, that makes computationally intractable for approximated methods and infeasible for exactly solutions. For that reason, testing environments in experiments is very limited. Moreover, mobile robots can only move on a topological map where only a few interesting points can be defined and a fully exploration would be computationally expensive as well. Another approach in this direction is given by [6], which presents a people classifier intended to be implemented to surveillance. Based on a system of classification states and uncertainties, it uses *Dynamic Bayesian Networks* [7] to determine the current belief state and consequently verify the characteristics of the involved goals.

In [8] and [9] are used aerial and ground vehicles to work cooperatively based on different sensed information. In [8] is proposed a probabilistic framework to cooperatively detect and track a target using a team composed by Unmanned Aerial Vehicles (UAV) and Unmanned Ground Vehicles (UGV).

The work presented by [10] shows a methodology for selection of regions of interest in images in order to extract the maximum of useful information about a scenario observed by a group of agents, the objective is to share only really important data. For this, the authors demonstrate the technique called *Semantic Stability*, that describes how to use the relationship between objects to differentiate with low or high quality of information, where objects with high semantic level represents information gain.

III. Problem Statement

Let be $\boldsymbol{\mathcal{W}} \subset \mathbb{R}^2$ a region bounded by a convex polygon, where the robot can sample and we define as the *robot workspace*.

In this environment, some kind of objects/entities can be identified by sampling (*eg.* trees, people, animals), or recognized by an expert (*eg.* people in danger, scared animals). Then, the n types of entities are defined as the set $\boldsymbol{\mathcal{T}} = \{t_1, t_2, ..., t_n\}$.

The aerial robot is able to capture multiple pictures in the workspace to identify objects, which we refer as evidences (a picture can contain multiple evidences). Those evidences are geo-localized using a localization system like a Global Positioning System device (GPS). Hence, at the detection stage, when the robot is sensing the environment, the set $\boldsymbol{\mathcal{E}} = \{e_1, e_2, ..., e_m\}$ of m evidences are sampled. Each evidence is composed of a tuple $e_i = (x_i, t_i)$ where x_i is the position of the evidence in the workspace $x_i \in \boldsymbol{\mathcal{W}}$ and the tuple $t_i \in \boldsymbol{\mathcal{T}}$ which identifies the object. Additionally, the knowledge of experts is also introduced as a *a-priori* knowledge in order to allow the system to infer about the latent entities that cannot be sampled by the robot. The association is made by a set of l rules $\boldsymbol{\mathcal{R}} = \{r_1, r_2, ..., r_l\}$, where each rule represents the association of types with the form

$$\boldsymbol{\mathcal{E}}' \xrightarrow[f]{} e_\prime \tag{1}$$

where f is a function which creates a relationship between existent evidences and the possibility of new ones. In other words, we take a subset of the evidences $\boldsymbol{\mathcal{E}}' \subseteq \boldsymbol{\mathcal{E}}$, apply an association function f to obtain a new evidence $e_\prime$ for each fired rule. The modeling of association functions and the relationship definition take us to the Problem 1.

Problem 1 (geo-localized rule definitions): How to define a way to create relationships based on evidence type and location in order to recognize latent variables by using the detections of the robot.

Then, designing a method to identify latent variables by combining robot data and knowledge experts, take us to the problem 2.

Problem 2 (information fusion): How to combine the robot data and the experts' knowledge in order to infer about latent variables.

Section IV proposes the rule definitions for Problem 1, and section V describes our proposal for Problem 2.

IV. Rule definitions for geo-localized environments

An efficient way for representing the experts' knowledge has been a challenge in the scientific community since the beginnings of the Artificial Intelligence field. One approach to define this kind of rules has been tackled by combining Conditional Probability Tables (CPT) and First Order logic [11], [12], [13], [14]. In [14] is presented a set of rules focused on inference for relational databases. In our approach, we want to extend these types of rules in order to work in geo-localized environments, where the association between two variables is made by the evidence type and Euclidean distance. Hence, the function f of the expression 1 is represented by the *Conditional Probability Table - CPT*, and we define $\boldsymbol{\mathcal{E}}$ as the set of all the inferred and detected evidences.

Each rule $r_i \in \boldsymbol{\mathcal{R}}$ is defined as the relationship between entities of any type $t_i \in \boldsymbol{\mathcal{T}}$. In the following, we present some type of rules to represent knowledge where the distance is an important factor to associate correlation between variables. These rules will be used to infer the hidden variables on the environment; using the *inference engine* described in section V.

1) Sensor rule: This rule describes the two variables of the *Bayesian* theorem (cause and effect). It is triggered when there is an evidence about the effect variable. The requirements are:

- *Cause variable:* hidden variable that a sensor tries to estimate, *e.g.* temperature.
- *Probability of the cause:* the possible states and its probabilities, *e.g.*: $hot = 0.7$, and $cold = 0.3$.

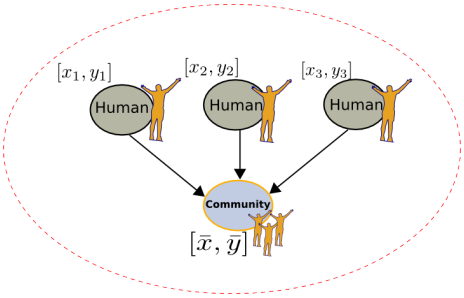

Fig. 1. Example for community rule. It groups three human evidences, located in positions $[x_1, y_1]$, $[x_2, y_2]$ and $[x_3, y_3]$, in order to create a community in the average location $[\bar{x}, \bar{y}]$.

- *Effect:* the generated variable based on the evidence or sensed value. *e.g.* for temperature: sensor measure $=$ *hot*.
- *Probability distribution function of the cause:* in the discrete case, it is the CPT. That is the sensor credibility based on previous data. It is fulfilled by the true positives, false positives, true negatives and false negatives. Table I shows an example for a temperature sensor.

	Hot	**Cold**
Sensor: hot	0.9	0.1
Sensor: cold	0.2	0.8

TABLE I

AN EXAMPLE OF A CPT FOR A TEMPERATURE SENSOR. IT REPRESENTS THE RELATIONSHIP BETWEEN THE REAL VALUES AND THE NOISY MEASUREMENTS.

2) Community rule: This rule is defined to group a variable in a point with its neighbors. Figure 1 illustrates an example, where multiple humans inside a circular area are grouped to create a community. The representation point of the community is the mean of the grouped points $[\bar{x}, \bar{y}]$. It requires to define:

- *Query variable:* variable in map to be grouped.
- *Community variable:* resultant variable.
- *Joint function:* to summarize the marginal probabilities of each point in the community. A *joint function* can be the mean, maximum, minimum, etc.
- *Maximum distance:* maximum distance between the points of the community.
- *Minimum number:* minimum number of elements to create a new community.

3) Query rule: For each trigger variable, a new query is made to search for evidences of a query variable around a distance (*e.g* find fire detections near to a human, where human is the trigger variable). All of the detections are mixed by a *joint function*. Figure 2 shows an example scenario of detecting a human in danger based on the presence of fire. The requirements to be defined by the expert are:

- *Cause variable:* this variable triggers the rule. When there is a new evidence about the cause variable, the rule will create a new inferred evidence based on the rule definition.

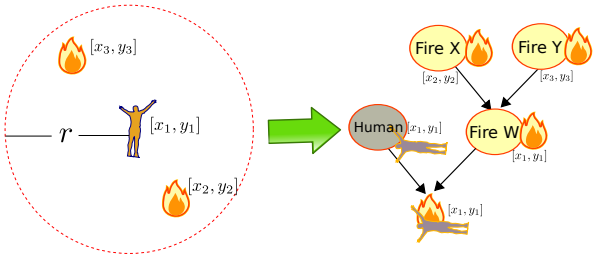

Fig. 2. An example of a query for a human in danger. It searches evidences of fire in a circular vicinity of radius r around the humans to infer the relationship between human and fire.

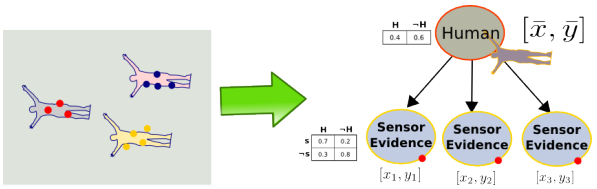

Fig. 3. Grouping multiple evidences that have similar locations in order to combine them for only one evidence.

- *Inferred variable:* variable to be inferred after the query.
- *Inferred CPT:* conditional probability table for inferred variable.
- *Query variable:* variable to query before inference.
- *Joint function:* is used to summarize the marginal probabilities of each point in the community. A *joint function* applies over a set of locations and can be the mean, maximum, minimum, etc.

4) Affinity rule: when there are detection uncertainty in the detection, the *sensor rule* cannot be applied directly, because many points of the same variable are spread around. Then applying the *sensor rule* would create multiple wrong beliefs of an unique evidence.

The *Affinity rule* groups similar points to infer about the same variable. It helps when the detector or the geolocalization system returns inexact position, and the measurements are located in very close places. The *affinity rule* applies a clustering method among all the points of the trigger variable.

Figure 3 shows an example of the *affinity rule*. Multiple humans were detected, but many detections in very close positions corresponding to the same human. Then, after clustering with the affinity propagation method [15], every point in a cluster infer about the same variable. The requirements for this rule are:

- *Minimum number of points:* minimum number to trigger an inference.
- *Cause:* cause variable or parent (hidden variable).
- *Effect:* effect variable or trigger variable.
- *Cause probabilities:* probability of the cause appears in the environment.
- *Effect CPT:* Conditional Probability Table for the effect variable.

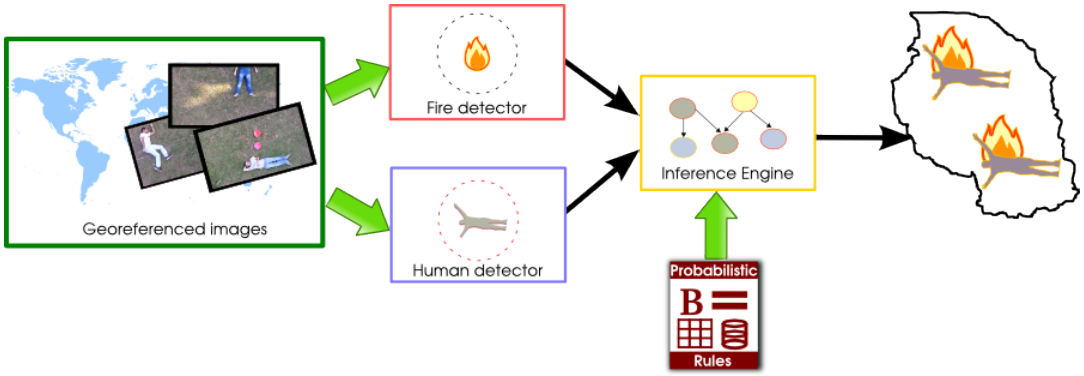

Fig. 4. Overview of our methodology for detection of latent variables by inference using geolocalized sensed data. We use the geolocalized images to detect variables of interest (*e.g.* fire and human detection). This information prorvides the evidences that the inference engine receives as input. The output represents the fusion between the experts' knowledge (defined by the probabilistic rules) and the evidences.

V. Inference Engine

The proposed approach is focused on fusing information from different sensors and *a-priori* expert's knowledge in order to infer about latent variables that can not be directly detected. There is no physical sensor to identify if a person is in danger, but an expert can give the key information to create inferences based on sensed data. Therefore, we use the knowledge representation of section IV to extract the users' expertise and combine it with sensed data.

The use of *Bayesian Networks* gives a way to create relationships about different variables. Some approaches create a strong assumption about a fixed structure for the *Bayesian network* [2] and others do not work on continuous space or geo-localized environments [6]. Our proposal is aimed to avoid those restrictions, we define a method to dynamically create the structure of the *Bayesian network* based on geo-localized information.

Figure 4 shows an illustration describing how our method works in the context of detecting people in danger after a catastrophe. The inference engine has as input the experts' knowledge defined by the set of rules $\boldsymbol{\mathcal{R}}$ and the sensors' detections as a set of evidences $\boldsymbol{\mathcal{E}}$. In this case, our sensors receive aerial photos with GPS coordinates and orientation. For visual sensors, each image must be processed to detect objects of interest. In the context of detecting humans in danger, we detect the entities people and fire.

The inference engine works in environments that can be defined in the Euclidean space $\mathbb{R}^2$. There is a set of classes or type of elements $\boldsymbol{\mathcal{T}}$ that can be detected by physical sensors. Then, the punctual evidences $e_i \in \boldsymbol{\mathcal{E}}$ are points in the environment, which were captured by a sensor.

After the rule definition stage, and new evidences appear, the algorithm 1 is executed in order to create the structure of the *Bayesian network*, represented as a directed graph, based on the rules in $\boldsymbol{\mathcal{R}}$ and the evidences $\boldsymbol{\mathcal{E}}$. This algorithm initially create a new vertex for each evidence (line 1), and a empty set for the edges (line 2). The loop from line 3 to 13 is repeated while no vertex or no edges appear after triggering a rule. In other words, when the set of vertices V is not empty, and the vertices as the edges are different from the computed in the last iteration. The set of rules $\boldsymbol{\mathcal{R}}$ is applied for the new and old vertices $V' \cup V$ (line 7) in oder to obtain new ones. Lines 8 and 9 apply the union for vertices V and edges E (it is important to remark that the sets do not contain duplicated elements). This process is repeated cyclically until no new nodes are created and finally return the bayesian network with the graph structure (V, E).

Algorithm 1: InferenceEngine(Evidences $\boldsymbol{\mathcal{E}}$, Rules $\boldsymbol{\mathcal{R}}$)

```
1  V ← E
2  E ← ∅
3  while V ≠ ∅ ∧ (V = V' ∩ V) ∧ (E = E' ∩ E) do
4      E' ← ∅
5      V' = ∅
6      foreach r_i ∈ R do
7          V_r, E_r ← r.infer(V' ∪ V)
8          V' ← V' ∪ V_r
9          E' ← E' ∪ E_r
10     end
11     E' ← E' ∪ E
12     V' ← V' ∪ V
13 end
14 return BN(V, E)
```

We would like to aware that this algorithm has a hazard; when a cyclic rule is defined, the algorithm will fall in a infinite loop. Hence, a rule definition must be validated *a priori* in order to avoid possible cycles.

VI. Experiments

In our experiments, we simulated a catastrophe in an outdoor environment, where there are some lied people and fire. The fire is represented by circular red cardboards. We used an aerial robot (*Asctec Quadcopter Hummingbird*), Figure 5, equipped with cellphone (Samsung Galaxy S4) to capture video at 30 FPS and resolution of 1280x720. The angle of view is perpendicular to the ground. The objective is to detect people in danger based on fire and human detections. In this context, a person is in danger when there is fire close to it.

Fig. 5. Quadcopter used to collect the visual data used in the experiments. It's used a cellphone camera to take pictures of the studied environment.

The quadcopter flies taking video of the environment at height of approximately 5m. The ground-truth was created by taking an aerial image at 30m that covers the hole environment. The experts' knowledge representation, used detectors, and the process to take the samples are described as following.

A. Expert's Knowledge

We defined five rules to represent the experts' knowledge. The experts defined the relationship between the variables and the CPT for each one.

1) *Fire detection rule:* a sensor rule with a trained sensor with the mentioned probabilities.
2) *Human detection rule:* an affinity rule with the mentioned probabilities for human detection.
3) *Human in danger:* a query rule to identify when there are fire and human in a range of 2.0m with a *joint function* maximum probability of having fire.
4) *People community:* a community rule to identify more than 2 humans in a radio of 2.5m.
5) *Community in danger:* similar to the *human in danger rule* but using a radius of 3m.

B. Detectors

In this work, we try infer if some person is in danger or not. For that reason, we define humans and fire as entities of interest. The entities in conjunction with the experts' knowledge can achieve the objective.

1) Fire Detector: Fire is one of the natural phenomena that can be considered dangerous in cases of burnings or explosions. We have represented fire in map as a red circle. It detects and highlights the position of red circles on a image, using a color filter, in the RGB color image space, and a *Hough transformation* [16] for detecting the circular shapes.

2) Human Detector: The people detector was based on a pedestrian detector, but with some additional treatment. In an aerial image, a person lied in the floor can be seen as a pedestrian with a rotated angle θ. Normally a pedestrian detector is trained to detect people totally perpendicular to the floor, *i.e.* an angle approximately of $\theta = \pi/2$ with respect to the image. Therefore, we used the same detector but trying with a set of angles Θ with the intuition that the optimal value θ for the detector will be approximately in the set Θ. As can be seen in Figure 6 for a Θ of nine different angles. We used the pedestrian detector *latent-svm* [17] with the trained *OpenCV* implementation.

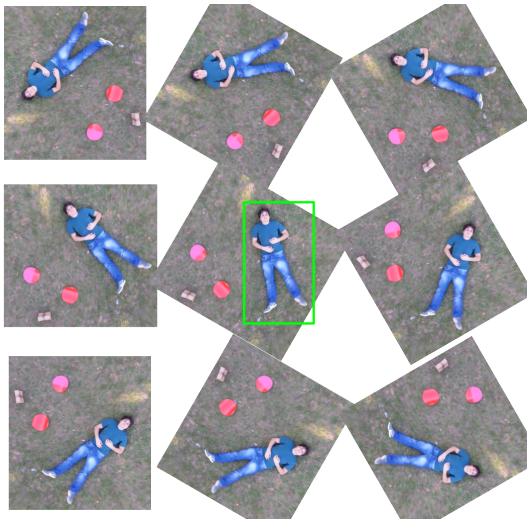

Fig. 6. Solution used to detect people lying on the ground using a pedestrian detector. The image is rotated into predetermined angles trying to find the closer angle where the person is in a pedestrian like position.

C. Sampling the environment

The images generated in the aerial capture are exemplified in Figure 7, that demonstrate people in different positions and rotations relative to the camera. It makes the simple pedestrian detector useless, because in rare cases the taken picture match with the optimal angle θ.

Figure 8 shows the raw sensed data. It has many false positives and lacks of human detection. After creating the *Bayesian network* and infering with it, the resultant processed information is shown in Figure 9. The pie chart represents the confidence of having a human in that position (green for true). We can see that there are two humans that were not detected. There were more false negatives than real detections in that area, then the result of the *Bayesian network* simply tries to create an estimation with the given data and the *a-priori* information. It could be improved by feeding the same points with a different detector in order to complement the information.

All the red circles were detected simulating fire. The result for humans in danger, and communities in danger is showed in Figure 10. The person with the green t-shirt were partially detected but it was not in danger because it did not have fire around. Only one community in danger where detected based on the given radio.

We studied an static scene where the robot were sampling in small areas. At the end we combine them using the inference engine and identify that: three people are in danger (true-positives), a partially detected human, a not detected person (false-negative) and a partially identified group of people by the community rule (As may be seen in Figure 10). Those results are not very optimistic, taking into account

Fig. 7. People lying on the floor with different positions closer to the fire marker (red circle). Images used as input for the detection algorithm.

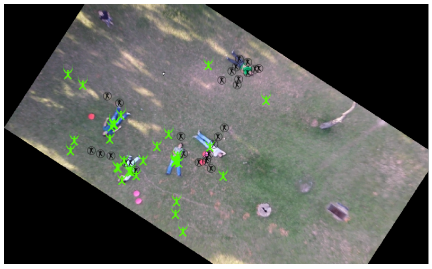

Fig. 8. Human detections. All the detections were geo-located and combined for this image. The green human shapes represent a detection and the black ones represent an area where no human was identified.

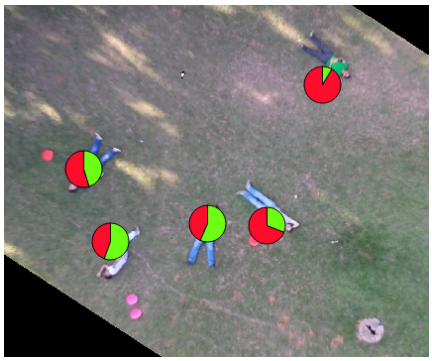

Fig. 9. Infered humans after the affinity rule. The pies represent the confidence about the detections, where green is for positive belief and red for negative.

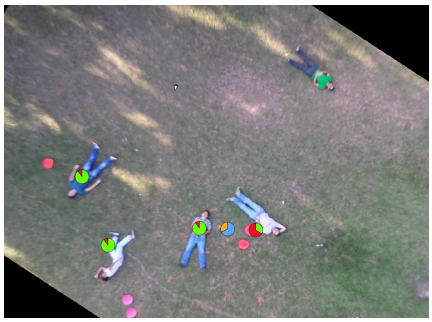

Fig. 10. Humans in danger (pie charts in green) and Communities in danger (pie charts in blue)

that a person was not identified, but if we see the very noisy input of Figure 8, the output of the proposal offers a relevant summary to act in a hazardous environment.

VII. Conclusions

We proposed an inference engine to detect latent variables fusing experts' knowledge and sampled data by physical sensors in geo-localized environments. We analyzsed our approach in a experiment that simulates a disaster and act as a *first response system*. The physical experiments showed that the information fusion improved the detections, not only to reduce the false-positives (given a very bad human detector with high probability for false-positives) but also to infer about latent variables.

We experimentally showed that the use of different sensor data and expert's knowledge based on Probabilistic rules and Bayesian Networks is a potential approach. It improves the accuracy about the detected elements and infer about latent variables that can not be sample directly with a physical sensor, for example the variable "human in danger".

This inference engine can be extended to more complex shapes, such as polygons or lines. That increment can help on detection of other types of cases, since it is based on rules previously constructed, making it a generic tool.

References

[1] R. Andrew Russell, D. Thiel, R. Deveza, and A. Mackay-Sim, "Robotic system to locate hazardous chemical leaks," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 556–561, 1995, cited By (since 1996)61.

[2] P. Drews Jr., J. Quintas, J. Dias, M. Andersson, J. Nygrds, and J. Rydell, "Crowd behavior analysis under cameras network fusion using probabilistic methods," *13th Conference on Information Fusion, Fusion 2010*, 2010.

[3] L. Rabiner and B.-H. Juang, "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.

[4] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[5] M. T. Spaan, "Cooperative active perception using pomdps," 2008.

[6] M. Spaan, T. Veiga, and P. Lima, "Active cooperative perception in network robot systems using pomdps," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 4800–4805, 2010.

[7] K. P. Murphy, "Dynamic bayesian networks," *Probabilistic Graphical Models, M. Jordan*, 2002.

[8] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 16–26, 2006.

[9] M. Garzón, J. Valente, D. Zapata, and A. Barrientos, "An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas," *Sensors*, vol. 13, no. 1, pp. 1247–1267, 2013.

[10] M. Rokunuzzaman, T. Umeda, K. Sekiyama, and T. Fukuda, "A region of interest (roi) sharing protocol for multirobot cooperation with distributed sensing based on semantic stability," 2014.

[11] K. B. Laskey, "Mebn: A language for first-order bayesian knowledge bases," *Artificial intelligence*, vol. 172, no. 2, pp. 140–178, 2008.

[12] V. Tresp, J. Hollatz, and S. Ahmad, "Representing probabilistic rules with networks of gaussian basis functions," *Machine Learning*, vol. 27, no. 2, pp. 173–200, 1997.

[13] S. Natarajan, P. Tadepalli, T. G. Dietterich, and A. Fern, "Learning first-order probabilistic models with combining rules," *Annals of Mathematics and Artificial Intelligence*, vol. 54, no. 1-3, pp. 223–256, 2008.

[14] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of relational structure," in *ICML*, vol. 1, 2001, pp. 170–177.

[15] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[16] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[17] C.-N. J. Yu and T. Joachims, "Learning structural svms with latent variables," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1169–1176.